大模型与标准文献知识库的融合应用探索

徐松林 [1]

[1]（云赛智联股份有限公司，上海 200233）

摘要：在人工智能与大数据技术背景下，利用大模型及构建标准文献知识库对于科研创新、知识挖掘和信息检索具有重要价值。标准文献知识库为各行业的规范化、标准化提供了坚实的支撑。本研究首先探讨了标准文献的现状，然后基于检索增强搭建大模型与标准文献知识库集成的框架，并提出各阶段增强优化探索。最后展望了未来的研究方向和应用前景。

关键词：大模型；标准文献；知识库；检索增强

中图分类号：TP3　　　　　　　　　文献标识码：A

Exploration of the Integration and Application of Large Model and Standard Literature Knowledge Base

Xu Songlin[1]

[1](INESA Intelligent Tech Inc., Shanghai 200233, China)

Abstract: In the context of artificial intelligence and big data technology, the use of large models and the construction of standard literature knowledge bases are of great value for scientific research innovation, knowledge mining, and information retrieval. The standard literature knowledge base provides solid support for the standardization and standardization of various industries. This study first explores the current status of standard literature, then builds a framework for integrating large models and standard literature knowledge bases based on retrieval enhancement, and proposes exploration of enhancement optimization in each stage. Finally, it looks forward to future research directions and application prospects.

Keywords: Large Language Models; Standard Literature; Knowledge Base; Retrieval-Augmented Generation

1　引言

我国一直以来都非常重视标准文献的建立，标准文献也是推动经济社会发展、提高产品质量、保障人民生活水平的重要手段。国家也在持续加大对标准文献的投入，不断完善各类标准体系，以推动经济社会持续健康发展。为各行各业的健康发展提供了有力保障。例如，在技术领域，标准对于技术创新、技术进步具有

重要的引导作用。在产品质量方面，标准是衡量产品优劣的重要依据。在安全生产方面，标准对于预防和减少事故发生具有重要意义。在环境保护领域，标准对于控制污染物排放、改善生态环境具有重要意义。

基于全国标准信息公共服务平台（https://std.samr.gov.cn/）公开的数据，截止2024 年 1 月 11 日，国家层面总共发布 68020 标准（不含食品安全、环境保护、工程建设方面的国家标准）、行业层面，包括现行标准 43693 部，即将实施标准1662 部，废止标准 22665 部。从这些数据来看，我国在标准工作方面的投入巨大，标准数量呈现出稳步上升的趋势。然而，在实际执行过程中，这些标准的实施效果并未如预期那样理想。

鉴于此，本研究提出一种融合大模型与知识库的方法。该方法涵盖标准文献知识提取、检索增强的 RAG 技术、各阶段的增强探索及评估等多个环节，旨在为该领域问题提供科学、有效的解决路径。

2 标准现状

除国家层面在制定标准外，我国各行业部门及地方政府也会按需制定相关标准。在 2021 年至 2023 年期间，各行业部门共发布 13007 部标准、废止 83 部标准，各地方政府共发布 28704 部标准、废止 1213 部标准，整体标准数量呈现逐年稳步增长趋势，具体数量详见表 1。

表 1 行业、地方发布及备案标准明细表（2021 年-2023 年）

2021 年标准情况					2022 年标准情况					2023 年标准情况				
月份	发布标准		废止标准		月份	发布标准		废止标准		月份	发布标准		废止标准	
	行业	地方	行业	地方		行业	地方	行业	地方		行业	地方	行业	地方
1 月	942	1480	0	1	1 月	342	1027	21	3	1 月	126	840	0	0
2 月	702	742	0	0	2 月	298	439	0	0	2 月	428	610	0	91
3 月	808	497	0	12	3 月	291	602	17	193	3 月	237	1029	1	0
4 月	470	438	0	0	4 月	685	648	0	1	4 月	754	830	0	0
5 月	360	449	0	9	5 月	136	655	10	4	5 月	130	461	0	2
6 月	613	599	0	0	6 月	352	583	0	1	6 月	453	851	30	4
7 月	225	541	0	7	7 月	198	672	4	0	7 月	223	1089	0	50
8 月	240	860	0	0	8 月	182	826	0	81	8 月	143	930	0	21
9 月	124	476	0	0	9 月	88	675	0	32	9 月	218	1133	0	8
10 月	39	473	0	73	10 月	280	952	0	0	10 月	244	961	0	1
11 月	235	954	0	58	11 月	626	423	0	69	11 月	947	1073	0	5
12 月	518	1079	0	2	12 月	34	1197	0	263	12 月	316	1610	0	222
合计	5276	8588	0	162	合计	3512	8699	52	647	合计	4219	11417	31	404

同时，针对上述标准名称进行深入剖析，梳理出各类别的词云图。在行业标准方面，核心词汇包括技术、方法、技术规范、测定、条件、通用、档案、液压、气象等。而在地方标准领域，关键词则表现为技术规程、规范、技术规范、生产、管理、服务、栽培、产品、地理、标志等。



行业备案标准词云图（2021-2023 年）



行业废止标准词云图（2021-2023 年）



地方备案标准词云图（2021-2023 年）



地方废止标准词云图（2021-2023 年）

图 1 行业、地方发布及备案标准词云图（2021 年-2023 年）

3 研究方法

3.1 概述

随着人工智能和机器学习的快速发展，大模型（Large Language Models，LLMs）成为数据科学领域的热门技术之一。大模型是指使用大规模数据进行训练的深度学习模型，具有强大的表征和推理能力，能够自动学习和理解自然语言的模式和语义，如 GPT-3、BERT 和 GPT-4 等[1-3]。随着 ChatGPT、文心一言等产品的相继发布，大模型在自然语言处理、文本生成、语义搜索等任务中展现出了令人瞩目的能力和效果[4-6]。知识图谱是一种结构化的知识表示方法，通过将实体、关系和属性组织成图形网络，为数据之间的联系和语义提供了更深入的理解[7]。

基于知识图谱的知识库具有丰富的语义表达和关联性，可以有效地描述现实世界中的各种事物及其关系。大模型非常适用于通用知识的生成，对于特定领域的专业知识还是表现欠缺，此时通过大模型结合知识库的方式可以有效解决此问题。

在专业知识引入层面，大模型引入知识库的策略主要为三种：

1、上下文引入（As Context）方法：将知识直接作为提示词的上下文，使每个问题都携带所需的全局知识。然而，此举将消耗更多 token，并受到大模型 token

的限制。

2、文本嵌入（Text Embeddings）：将知识进行向量化并存储在向量数据库中。这种方法在模型中引入了短期记忆，对于针对性问题较为有效。然而，由于嵌入的知识量需要受限，否则可能对模型产生较大噪声。

3、微调（Fine Tuning）：将知识库处理为微调数据集，然后对模型进行微调。微调后的模型具备长期事实记忆特点，回答问题更具条理性。但针对特定知识的回复能力略逊于文本嵌入。

在文本生成的增强方面，大模型引入知识库主要做两部分的优化：

1、对生成文本进行知识补充与校正：当大模型生成的文本不完整或存在错误时，可借助知识库进行补充或修正。根据知识库中的真实事实，对大模型的生成文本进行修改或补充，以提升回答的准确性。例如，若大型模型在回答某一问题时遗漏了关键细节，可从知识库中提取相关内容并融入回答之中。

2、运用知识库中的具体实体对生成文本进行替换：大模型生成的文本中可能含有通用的实体描述，如“小米创始人”。通过知识库中的具体实体替换这些通用描述，使回答更具针对性和准确性，如将“小米创始人”替换为“雷军”。

3.2 标准文献图谱提取

标准化是为了建立最佳秩序、促进共同效益而开展的制定并应用标准的活动。为了保证标准化活动有序开展，促进标准化目标和效益的实现，对标准化活动本身确立规则已经成为国内外各类标准化机构开展标准化活动的首要任务[8]。

《标准化工作导则 第1部分:标准化文件的结构和起草规则（GB/T 1.1-2020）》是我国标准化领域的重要标准，规范标准的编写和表述，提高标准的质量和可读性。此标准适用于各种领域的一般技术文档、管理文档和行政文件等，对于文档构成、内容格式、编写规则和排版规范都做了明确规定，对大量非结构化的标准文献开展智能化、自动化解析打下了坚实的基础。

构建一套规范且系统的标准文献知识图谱，并通过不断的完善，逐步形成一个全面的知识库，是开展知识检索、分析、推理、校对等各项工作的基础和先决条件。一要逐步建立全面准确的与大数据、数据经济及数据要素紧密相关的标准文献体系，深入解读国家、行业及地方在相关领域所提出的具体规范和明确要求。二要系统梳理标准文献之间的相互关联，进而构建出标准文献的知识图谱（图2：

标准文献知识图谱）。三是针对某些关键热词，如“数据要素×”和“新质生产力”等进行深入分析，以预测和评估其未来的标准内容及其可能产生的关联影响，为制定相应的应对策略提供有力支持。

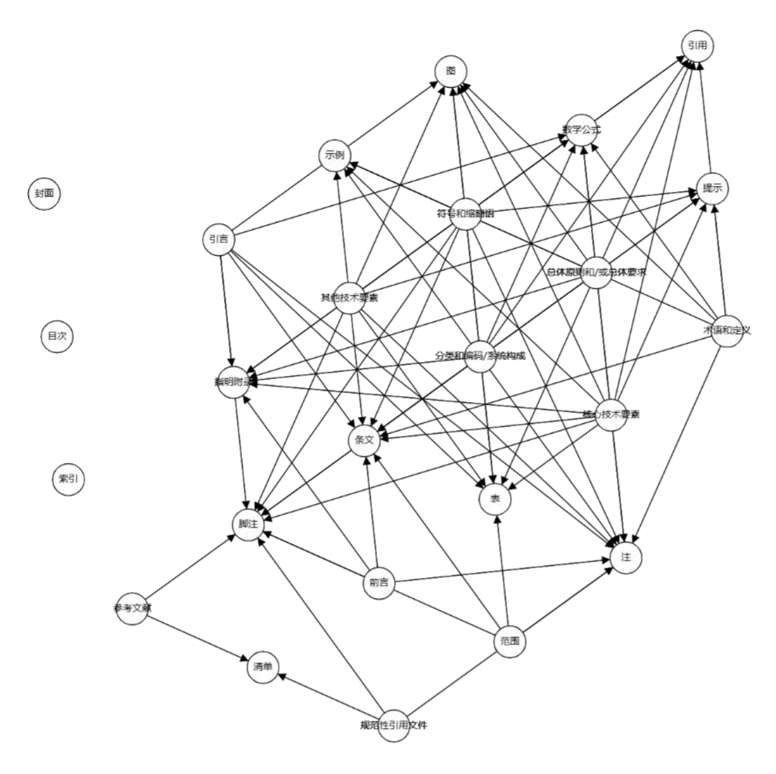


图 2 标准文献图谱

在本研究中选取了多部与大数据、数据经济及数据要素相关的标准文献，包括：《公共数据中台建设规范》、《区级大数据资源平台建设指南》、《公共数据共享交换工作规范》、《公共数据运营服务实施指南》和《公共数据治理能力评估规范》等。通常情况下，标准文献以 PDF 文件格式存储，因此需要采用“自动提取与人工审核”相结合的方式，对这些文献中的文本和表格等内容按照上述知识图谱的三元组信息进行提取。首先通过运用 PDFPlumber、pyMuPDF 等框架，实现对标准文献的自动化解析，涵盖术语、定义、表格、图表、正文解析、附录解析等各个方面。针对含有表格的标准，额外采用 Camelot 工具提取其中的表格。接着，对提取所得内容进行去噪、分词、词干提取等处理，以提升文本质量。最后对上述自动提取的内容进行人工审核，确保符合模型质量要求。

3.3 大模型与知识库集成

检索增强 RAG（Retrieval-Augmented Generation）是一种结合检索和生成的方法，用于实现大模型与知识库的集成。这种方法通过在大模型中引入外部知识

库的信息，提高了模型的生成能力和准确性。大模型与知识库集成的关键步骤主要包括语料预处理、向量数据生成以及检索增强 RAG 等核心环节。

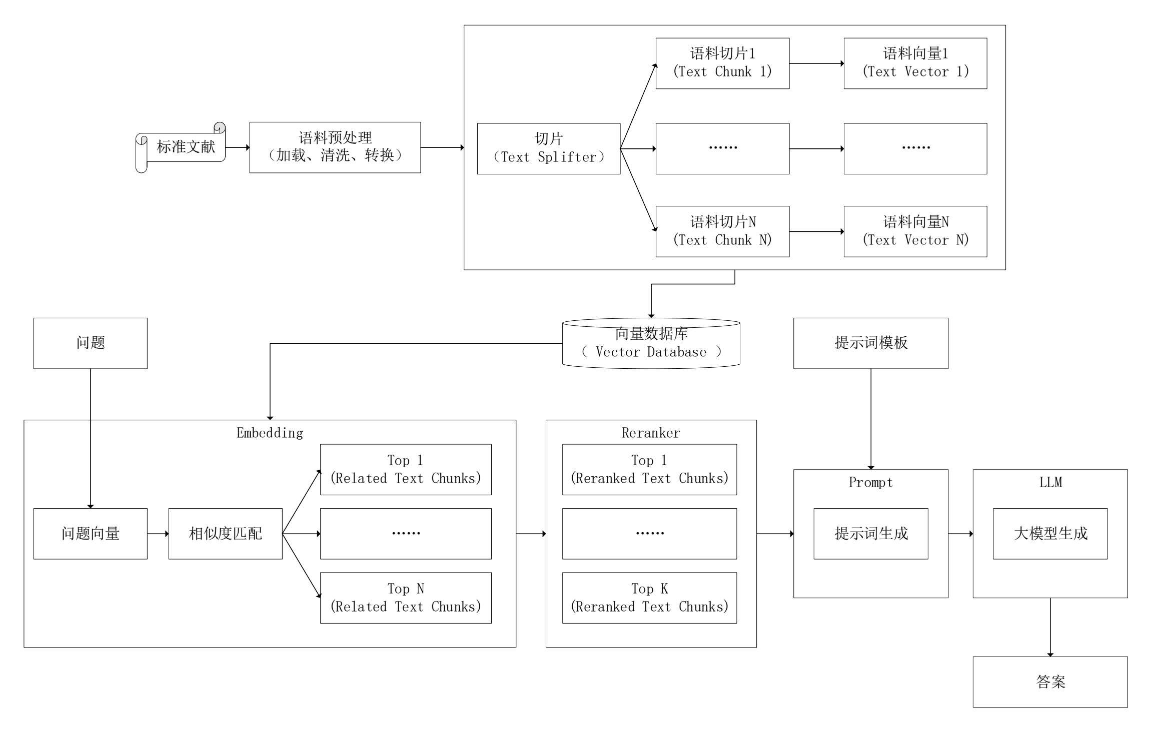


图 3 大模型与知识库集成架构图

在语料预处理及向量数据生成环节，主要分为以下三个步骤。

首先，进行标准文献收集及语料预处理：根据前述选定的标准文献文本，剔除停用词、填充词、重复词、无关词、拼写错误等可能干扰内容的词语，从而形成预处理后的语料。

其次，实施语料向量生成：借助 Embedding 模型（如智源研究院的 bge-large-zh 模型）构建向量数据库。将上述语料分割为小文本片段（Chunk），并分别将其映射为低维稠密向量。

接着，进行向量存储：将生成的向量存入向量数据库。向量数据库作为一种高效存储和检索向量数据的方法，有助于更好地处理和分析高维数据。本研究选用开源的 Milvus 作为本地向量数据库。

算法 1：向量生成	
1.语料预处理	corpus = read_literature(selected_literature) # 读取选定文献 preprocessed_corpus = preprocess(corpus) # 剔除停用词、填充词、重复词、无关词、拼写错误等
2.语料向量生成	embedding_model = load_embedding_model() # 初始化 Embedding 模型 chunks = split_into_chunks(preprocessed_corpus) # 分割语料为小文本片段（Chunk） for chunk in chunks: vector = embedding_model.encode(chunk) # 生成向量

	vectors.append(vector)
3.进行向量存储	database = connect_to_database() # 连接 Milvus 向量数据库 for vector in vectors: database.insert(vector) database.close()

当用户提出任一问题时，需要经过以下一系列步骤：

首先，查询向量生成：借助 Embedding 模型将问题转换为向量形式。

接下来，计算向量相似度：将查询向量与数据库中的每个向量进行相似度评估，可采用余弦相似度、欧氏距离等度量方法。根据相似度得分对查询结果进行排序、筛选，以获取最相似的向量，最终呈现 Top K 个相关文本片段。

然后，对文本进行精排：选用 Reranker 模型（如智源研究院的 bge-reranker-large）对文本片段进行重新排序。此步骤并非必需，需根据业务场景进行测试后确定。

接着，提示词工程：基于精排后的文本及提示词模板，生成问题对应的提示词。

最后，答案生成：依据提示词并利用 LLM 模型（如 ChatGLM3-6B）来生成答案。ChatGLM3 是智谱 AI 和清华大学 KEG 实验室联合发布的对话预训练模型[9]。

算法 2：大模型与知识库集成	

1.查询向量生成	embedding_model = load_embedding_model() # 初始化 Embedding 模型
	query_vector = embedding_model.encode(question) # 将问题转换为向量形式

2.计算向量相似度并获取文本片段	vector_database = connect_to_database() # 连接 Milvus 向量数据库
	for index, db_vector in enumerate(vector_database): # 遍历数据库中的每个向量
	  score = similarity_metric(query_vector, db_vector) # 计算查询向量与数据库内向量的相似度
	  similarity_scores.append((score, index))
	# 根据相似度得分排序并筛选 Top K 个
	sorted_scores = sorted(similarity_scores, reverse=True)
	top_k_indices = [score_index[1] for score_index in sorted_scores[:k]]
	for index in top_k_indices: # 获取 Top K 个文本片段
	  similar_texts.append(vector_database.get_text_by_index(index))

3.精排	reranker_model = load_reranker_model() # 初始化 Reranker 模型
	reranked_texts = reranker_model.rerank(similar_texts) # 使用 Reranker 模型对文本片段进行重新排序

4.提示词工程	hint_template = ********* # 提示词模板
	for text in reranked_texts: # 基于精排后的文本及提示词模板生成提示词
	  hint = hint_template.format(text=text)
	  hint_words.append(hint)

5.答案生成	llm_model = load_llm_model()　# 初始化 LLM 模型 answer = llm_model.generate_answer(hint_words)　# 利用 LLM 模型生成答案

3.4 各阶段增强优化探索

经过上述集成流程，从知识库中检索了相关内容，并将其传递至大模型进行结果生成。然而，在模型验证阶段，发现结果中存在部分与问题不相关或价值较低的信息，导致最终输出的结果未能达到预期标准。此外，当回答未能满足用户期望时，缺乏有效的调整手段。

鉴于当前现状，必须对问题进行深入迭代，或对内容进行审慎筛选与排序，或对解决方案进行必要改进。根据 RAG 的进一步增强分成多种类型，包括输入增强、检索器增强、生成器增强、结果增强以及 RAG 流程增强五种，详见表 2。

表 2 各阶段的增强方式

阶段	增强方式	描述
输入增强	查询转换	通过修改输入查询来增强检索结果，例如使用 Query2doc 和 HyDE 技术生成伪文档作为检索的键。
	数据增强★	在检索前对数据进行预先完善，如去除不相关信息、消除歧义、更新等，以提高 RAG 系统的性能。
检索器增强	递归检索	通过拆分查询并执行多次搜索来检索更多高质量的内容。
	块优化	通过调整检索文本块的大小来获得更好的检索结果，如句子窗口检索。
	微调向量嵌入模型★	使用高质量的领域数据或任务相关数据对检索器进行微调，以提高其性能。
	混合检索	同时使用多种类型的检索方法，如结合密集和稀疏检索器。
	重新排序★	对检索到的内容进行重新排序，以增加多样性和改善结果。
	元数据过滤	使用元数据（如时间、目的等）过滤检索到的文档，以获得更好的结果。
生成器增强	提示工程★	通过设计有效的提示来提高语言模型输出的质量。
	微调解码器	在生成过程中增加额外的控制，如调整解码器的温度或限制输出词汇表。
	微调生成器	对生成器进行微调，以增强模型的领域知识或与检索对象更好地匹配。
结果增强	重写输出	对 RAG 的最终结果进行修改，以适应特定的上下文或提高质量。
RAG 流程增强	自适应检索★	根据模型的置信度或其他指标决定是否进行检索。
	迭代 RAG	通过迭代的方式协同检索和生成，以提高生成内容的质量。
备注：带“★”标识的增强方式表示在本研究中采用。		

为科学评估模型性能，本研究选用了检索增强生成基准（Retrieval-Augmented Generation Benchmark，简称 RGB）作为测试标准[11]。该基准围绕噪声鲁棒性、负面拒绝、信息整合及反事实鲁棒性四大核心维度展开，旨在全面、客观地反映模型在实际应用中的性能表现。其中噪声鲁棒性评估大模型在包含噪声文档中提取有用信息的能力。负面拒绝评估大模型在检索到的文档中不存在所需知识时拒绝回答问题的能力。信息整合评估是否能够回答需要从多个文档中整合信息的复杂问题。反事实鲁棒性评估大模型在检索到的文档中存在已知事实错

误时，是否能够识别风险，并在通过指令给出潜在风险警告时，优先考虑自己的现有知识而不是检索到的信息[11]。

本研究经过对多种增强模式进行严谨验证与筛选，最终采用了“数据增强”、“微调向量嵌入模型”、“重新排序”、“提示工程”和“自适应检索”等策略。其中“自适应检索”策略致力于通过持续的优化过程，以提高最终结果的精确性和有效性[12]。当输出结果未能满足预期时，研究将回溯至关键节点，实施针对性优化措施。随后，研究将重新执行流程，直至满足既定条件，确保最终呈现一个令人满意的优质输出[13]。

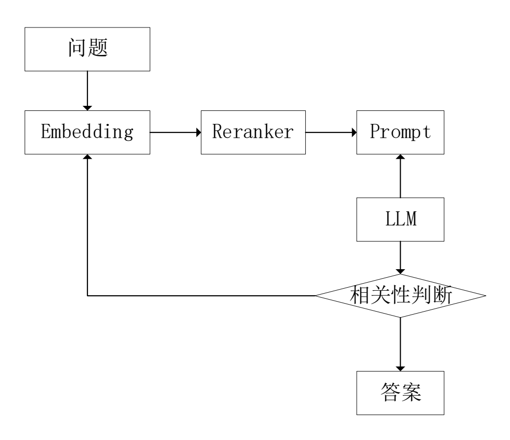


图 4 自适应检索

首先，通过问题、向量数据库以及提示词模型等协同进行检索，以生成初步的结果。随后，对生成的结果进行相关性判断。若结果的相关性低于相关性阈值，则将其视为负面答案，并据此重新调整提示词，再次传入大型模型以生成新的结果。这一过程将持续进行，直至生成的答案满足相关性要求。

算法 3：自反馈算法	

1. 提示词生成	query_vector = embedding_model.encode(question) # 查询向量生成
	similar_texts = get_top_k_similar_texts(query_vector, vector_database, similarity_metric, k=10) # 计算向量相似度并获取片段
	reranked_texts = reranker_model.rerank(similar_texts) # 精排
	hint_words = generate_hint_words(reranked_texts,hint_template) # 提示词工程

2.生成结果	llm_model = load_llm_model() # 初始化 LLM 模型
	answer = llm_model.generate_answer(hint_words) # 利用 LLM 模型生成答案

3.反馈结果	while not is_answer_relevant(answer, question):
	  hint_words = generate_hint_words(hint_words,answer) # 基于答案重新生成提示词，答案作为负面条件
	  answer = generate_answer(question, hint_words) # 重新利用 LLM 模型生成答案
	return answer

函数 1：判断答案	def is_answer_relevant(answer, question, relevance_threshold=0.8):

的相关性	# 调用相关性判断模型或函数
	relevance_score = relevance_model.score(answer, question)
	return relevance_score >= relevance_threshold

本研究以《公共数据治理能力评估规范》标准为例，对模型优化前后的输出结果进行了详细记录与对比分析，具体如表 3 所示。通过对比分析发现，针对问题一，优化前后的模型输出结果大致相同，均符合相关标准与要求。然而，在问题二的回答方面，经过优化后的模型表现出显著的提升，其效果明显优于优化前的模型，这充分说明了优化工作的重要性和必要性。因此，在大模型与知识库结合的过程中，引入自适应检索等策略机制，能够显著提高检索和生成的质量，从而进一步增强答案的精确性和相关性。

表 3 模型优化前后对比

问题	优化前	优化后
问题一：指标体系包括哪几条？	公共数据治理能力评估规范的指标体系是一个全面且多维度的框架......包括以下几个方面： 1.数据规划：评估组织是否制定了明确的数据规划策略，包括数据收集、存储、处理和应用等方面的规划。 2.组织管理：考察组织在数据治理方面的组织架构、人员配置以及职责划分是否合理有效。	公共数据治理能力评估规范的指标体系包括以下几个方面： 1.数据规划：包括目标规划、规划实施、规划评估三个能力项。 2.组织管理：包括组织架构、组织效能两个能力项。 ......
问题二：能力等级怎么划分？	公共数据治理能力评估规范中的能力等级划分通常基于组织在数据治理各个方面的表现和成熟度。...... 1.初始级（基础级）：组织对数据治理有基本的认识和了解，但尚未形成完整的数据治理体系...... 2.受管理级（发展级）：组织已经建立了初步的数据治理框架，并开始实施一些基本的数据管理制度和规范...... 3.标准级（优化级）：组织已经建立了完善的数据治理体系，并全面实施了一系列成熟的数据管理制度和规范......	能力等级的划分遵循以下原则和方法： 1.指标等级划分：.....能力等级应设置为 M 级，建议可分为 3 级或 5 级，不超过 9 级。 2.权重设计：......可以采用主观经验、主次指标排队分类法、层次分析法、专家打分法等方法，...... 3.治理能力等级：公共数据治理能力等级与评估指标的等级划分层级一致。...... 4.计算方法：能力项等级分数通过对评估指标分值进行加权求和的方法计算。......

4 结束语

本研究精心选取多部具有典型性的标准文献作为深入探讨的对象。通过充分利用 Embeddings、LLM、Rerander 等大模型的语义解析能力与知识库的资讯整合功能，全面细致地剖析了这些标准文献。同时，本研究还创新性地引入了“数据增强”和“自适应检索”等策略，对模型进行了精细化的优化。这些举措使得模型在标准文献领域取得了显著的提升和突破。

大模型最典型的应用就是写作生成。根据主题、关键词要求，大模型可以“写

作”故事、小说、诗歌、信件、新闻报道、时事评论、论文大纲等；进行文字修改和润色，如语法更正、文本翻译、关键字提取[10]。本研究具有广泛的领域适用性，在知识挖掘与检索方面，能够高效地对标准文献知识库进行检索、理解和推理，进而显著提高科研效率。在辅助写作与评审方面，针对论文写作、审稿和修订过程中涉及的标准文献，实现学术语义生成、校对核对等多种功能。举例来说，当面临如“在数据经济领域，常见的标准文献有哪些？”或“请概括某标准的核心理念及主要内容”等问题时，本研究能够自动生成准确、全面的答案，为用户提供有力的支持与帮助。

然而，研究之路并无止境。下一步，将拓展研究范围，不再局限于标准文献，而是将政策文件、新闻稿、电子公文等各类语料素材纳入研究范畴，进一步扩大大模型在政务信息处理领域的应用范围。此外，除本研究的方案外，还将深入研究微调技术，以构建政务领域垂直大模型。同时，通过对比两者在政务领域的优劣势，期望能确定大模型在政务场景下的推荐方案。

总之，基于大模型与知识库的标准文献研究具有巨大的发展潜力。通过不断的探索与实践，能在政务领域取得更为卓越的成果，为决策者提供有力的信息支持。

参考文献：

[1] 赵朝阳,朱贵波,王金桥.ChatGPT 给语言大模型带来的启示和多模态大模型新的发展思路[J].数据分析与知识发现,2023,7(03):26-35.

[2] 罗江华,张玉柳.多模态大模型驱动的学科知识图谱进化及教育应用[J].现代教育技术,2023,33(12):76-88.

[3] 朱光辉,王喜文. ChatGPT 的运行模式、关键技术及未来图景[J]. 新疆师范大学学报(哲学社会科学版), 2023,44(4):113-122.

[4] 邱锡鹏. 解剖大型语言模型：原理、应用及影响[J]. 探索与争鸣, 2023(5):10-12.

[5] 郭全中,杨元昭. 大语言模型的技术特征及市场新进展[J]. 中国传媒科技, 2023(5):159-160.

[6] 王静仪. 大型语言模型技术的影响、挑战与应对策略[J]. 华东科技, 2023(6):96-98.

[7] 王明皓,殷涛,杨洪杰,等. 知识图谱和大模型技术发展与应用[J]. 网络安全与数据治理, 2023, 42(S1): 126-131.

[8] 国家市场监督管理总局 国家标准化管理委员会. GB/T 1.1-2020 标准化工作导则 第 1 部分: 标准化文件的结构和起草规则[S]. 北京:中国标准出版社, 2020.

[9] 孙柏林. 大模型评述[J]. 计算机仿真, 2024,41(01):1-7+24.

[10] 支振锋.生成式人工智能大模型的信息内容治理[J].政法论坛,2023,41(04):34-48.

[11] Chen, Jiawei, Hongyu Lin, Xianpei Han and Le Sun. “Benchmarking Large Language Models in Retrieval-Augmented Generation.”[EB], [2023-12-20]. https://arxiv.org/abs/2309.01431

[12] Self-Reflective RAG with LangGraph[EB], [2024-02-07]. https://blog.langchain.dev/agentic-rag-with-langgraph/.

[13] Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil and Hannaneh Hajishirzi. “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection.”[EB], [2023-10-17]. https://arxiv.org/abs/2310.11511